

**UTILITY
PATENT APPLICATION
TRANSMITTAL**

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.

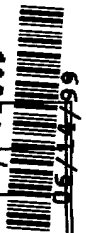
1960.125

First Named Inventor or Application Identifier

STEPHEN MARYKA, ET AL.

Express Mail Label No.

10551 U.S. PTO
09/332069



APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO:

Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

- ☒ Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)
2. ☒ Specification Total Pages
3. ☒ Drawing(s) (35 USC 113) Total Sheets
4. ☒ Oath or Declaration Total Pages
- a. ☐ Newly executed (original or copy)
- b. ☐ Unexecuted for information purposes
- c. ☐ Copy from a prior application (37 CFR 1.63(d))
(for continuation/divisional with Box 17 completed)
[Note Box 5 below]
- i. ☐ **DELETION OF INVENTOR(S)**
Signed Statement attached deleting
inventor(s) named in the prior application,
see 37 CFR 1.63(d)(2) and 1.33(b).
5. ☐ Incorporation By Reference (useable if Box 4c is checked)
The entire disclosure of the prior application, from which a copy of
the oath or declaration is supplied under Box 4c, is considered as
being part of the disclosure of the accompanying application and is
hereby incorporated by reference therein.

6. ☐ Microfiche Computer Program (Appendix)
7. Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)
- a. ☐ Computer Readable Copy
- b. ☐ Paper Copy (identical to computer copy)
- c. ☐ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. ☒ Assignment Papers (cover sheet & document(s))
9. ☐ 37 CFR 3.73(b) Statement (when there is an assignee) ☒ Power of Attorney
10. ☐ English Translation Document (if applicable)
11. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)
14. ☒ Small Entity Statement(s) ☐ Statement filed in prior application
Status still proper and desired
15. ☐ Certified Copy of Priority Document(s)
(if foreign priority is claimed)
16. ☐ Other: _____

17. If a CONTINUING APPLICATION, check appropriate box and supply the requisite information:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No. ____/____

18. CORRESPONDENCE ADDRESS

☒ Customer Number or Bar Code Label

05514
(Insert Customer No. or Attach bar code label here)

or ☐ Correspondence address below

NAME

Address

City

State

Zip Code

Country

Telephone

Fax



CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
	TOTAL CLAIMS (37 CFR 1.16(c))	20 -20 =	0	X \$ 18.00 =	\$00.00
	INDEPENDENT CLAIMS (37 cfr 1.16(b))	4 -3 =	1	X \$ 78.00 =	\$78.00
	MULTIPLE DEPENDENT CLAIMS (if applicable) (37 CFR 1.16(d))			\$260.00 =	\$00.00
				BASIC FEE (37 CFR 1.16(a))	\$760.00
	Total of above Calculations =				\$838.00
	Reduction by 50% for filing by small entity (Note 37 CFR 1.9, 1.27, 1.28).				\$419.00
	TOTAL =				\$419.00

19. Small entity status

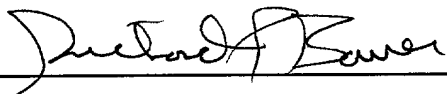
- a. ☐ A Small entity statement is enclosed
- b. ☐ A small entity statement was filed in the prior nonprovisional application and such status is still proper and desired.
- c. ☐ Is no longer claimed.

20. ☐ A check in the amount of \$ _____ to cover the filing fee is enclosed.

21. ☐ A check in the amount of \$ _____ to cover the recordal fee is enclosed.

22. The Commissioner is hereby authorized to credit overpayments or charge the following fees to Deposit Account No. 06-1205:

- a. ☒ Fees required under 37 CFR 1.16.
- b. ☐ Fees required under 37 CFR 1.17.
- c. ☐ Fees required under 37 CFR 1.18.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED	
NAME	RICHARD P. BAUER, REG. NO. 31,588
SIGNATURE	
DATE	JUNE 14, 1999

Applicant or Patentee: AUDES TECHNOLOGY INC. Attorney's Docket No.: T8-465038US
Serial or Patent No.: N/A
Filed or Issued: N/A
For: METHOD AND APPARATUS FOR INCREMENTAL DOWNLOAD FROM SERVER TO CLIENT

VERIFIED STATEMENT (DECLARATION) CLAIMING
SMALL ENTITY STATUS
(37 CFR 1.9(f) and 1.27(b))
INDEPENDENT INVENTOR

As a below named inventor, I hereby declare that I qualify as an independent inventor as defined in 37 CFR 1.9(c) for purposes of paying reduced fees under section 41(a) and (b) of Title 35, United States Code, to the Patent and Trademark Office with regard to the invention entitled METHOD AND APPARATUS FOR INCREMENTAL DOWNLOAD FROM SERVER TO CLIENT described in

- ☒ the specification filed herewith
☐ application serial no. _____, filed ____
☐ patent no. _____, issued _____

I have not assigned, granted, conveyed or licensed and am under no obligation under contract or law to assign, grant, convey or license, any rights in the invention to any person who could not be classified as an independent inventor under 37 CFR 1.9(c) if that person had made the invention, or to any concern which would not qualify as a small business concern under 37 CFR 1.9(d) or a nonprofit organization under 37 CFR 1.9(e).

Each person, concern or organization to which I have assigned, granted, conveyed, or licensed or am under an obligation under contractor law to assign, grant, convey, or license any rights in the invention is listed below:

- ☐ no such person, concern, or organization
☒ persons, concerns or organizations listed below*

*NOTE: Separate verified statements are required from each named person, concern or organization having rights to the invention averring to their status as small entities. (37 CFR 1.27)

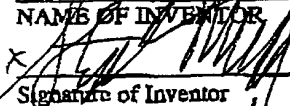
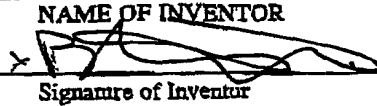
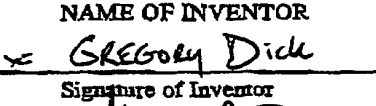
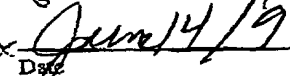
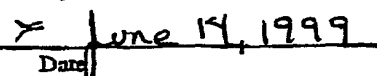
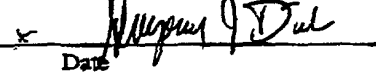
FULL NAME AUDES TECHNOLOGY INC.
ADDRESS 6815 8TH STREET, N.E., SUITE 180, CALGARY, ALBERTA T2E 7H7 CANADA
☐ Individual ☒ Small Business Concern ☐ Non-Profit Organization

FULL NAME _____
ADDRESS _____
☐ Individual ☐ Small Business Concern ☐ Non-Profit Organization

FULL NAME _____
ADDRESS _____
☐ Individual ☒ Small Business Concern ☐ Non-Profit Organization

I acknowledge the duty to file, in this application or patent, notification of any change in status resulting in loss of entitlement to small entity status prior to paying, or at the time of paying, the earliest of the issue fee or any maintenance fee due after the date on which status as a small entity is no longer appropriate. (37 CFR 1.28(b))

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that wilful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such wilful false statements may jeopardize the validity of the application, any patent issuing thereon, or any patent to which this verified statement is directed.

STEPHEN MARYKA NAME OF INVENTOR	BERTRAND MICHAUD NAME OF INVENTOR	GREG DICK NAME OF INVENTOR
 Signature of Inventor	 Signature of Inventor	 Signature of Inventor
 Date	 Date	 Date

JUNE 14 1999

METHOD AND APPARATUS FOR INCREMENTAL DOWNLOAD
FROM SERVER TO CLIENT

Field of the Invention

5 The present invention relates generally to file transfers over a communication medium and more particularly relates to a method and system for transferring objects from one computer to another.

Background of the Invention

10 The development of network computing ("NC") technology has spawned the development of several intelligent devices, ranging from simple thin-client desk-top computers, to internet-aware screen phones, mobile phones, personal digital assistants ("PDAs"), public kiosks, smart-card based banking devices, etc. The Java computer language has been an important feature of this technological development, as it provides a
15 "Write Once, Run Anywhere" platform which is an effective way to transfer an application from a server to a device for local execution on the device. Thus, Java provides the ability to transparently deliver, via a communications mechanism such as a general purpose network or a special purpose communications port, software to a variety of devices having different hardware platforms and has become a standard language for internet applications.

20 Additional transfer functionality is provided in the more recent *Java Dynamic Management Kit*, and discussed in *Java Dynamic Management Kit 3.0 Programming Guide*, 1998 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303 U.S.A. ("*JDMK*").

25 While Java and the *JDMK* provide effective ways to pass or transfer a software application to a device, they do not provide solutions to certain problems that can arise with file transfers. For example, existing solutions do not provide a robust recovery mechanism once power is restored, should power to the device be interrupted during a transfer. Further, where the communication medium has reduced bandwidth, it can be desirable to only download the components of the application which are not already

resident on the device. In addition, Java and the *JDMK* do not provide an infrastructure for managing the persistent storage of Java objects.

Summary of the Invention

5 It is an object of the present invention to provide a novel method and system for transferring objects between computers which obviates or mitigates at least one of the disadvantages of the prior art.

The present invention provides a method and system for transferring objects between computers, especially a server and a device in a Java environment. A server
10 connected to the device interrogates the device to determine whether the device has the available resources to accept a download, and to determine if any of objects included in the download are already resident on the device. Using a gateway object on the device, and a gateway managed-object on the server, the server passes an archive of the necessary objects, specifically those not presently available on the device, to the gateway via the
15 gateway managed-object. The gateway then instructs a persistent storage repository on the device to commit the archive to persistent storage.

A flag is set at the start of the archive-committing process, such that if the device loses power during the archive-committing process, upon re-initialization of the device, the device can determine that the archive-committing process failed and will instruct the
20 persistent storage to free-up storage area used during the archive-committing process. The transferal of objects can then be recommenced once the server and device reestablish communication.

The present invention can provide an effective means to ensure a stable recovery of the device upon re-initialization, should the file transfer and/or archive-commit process fail
25 due to a power failure to the device. Further, the invention can increase efficient use of device resources by not loading redundant components on the device and can reduce bandwidth requirements and/or download times as redundant information is not transferred.

Brief Description of the Drawings

The present invention will now be explained, by way of example only, with reference to certain embodiments and the attached Figures in which:

Figure 1 is a schematic diagram of a system for transferring objects between two computers in accordance with an embodiment of the present invention;

Figure 2 is a flow-chart of a method for transferring objects between two computers in accordance with an embodiment of the present invention;

Figure 3 is a schematic diagram of the system of Figure 1 showing the performance of a step of the method of Figure 2;

Figure 4 is a schematic diagram of the system of Figure 1 showing the performance of a step of the method of Figure 2;

Figure 5 is a schematic diagram of the system of Figure 1 showing the performance of a step of the method of Figure 2;

Figure 6 is a schematic diagram of the system of Figure 1 showing the performance of a step of the method of Figure 2; and

Figure 7 is a schematic diagram of the system of Figure 1 showing the performance of a step of the method of Figure 2.

Detailed Description of the Invention

Referring now to Figure 1, a system for transferring objects between two computers is indicated generally at 20. System 20 includes two computers, a device 22 and a server 24. Device 22 is any intelligent device as will occur to those of skill in the art, and examples of such devices include a 'thin-client terminal', an internet-aware screen phone, mobile phone, personal digital assistant ("PDA"), public kiosk and/or smart-card based banking device, etc. and is operable to execute software applications created in a language such as Java.

In a present embodiment of the invention, device 22 is Java-based and has a basic set of hardware resources including a central processing unit (not shown), and a persistent storage means (not shown) such as EEPROM, flash memory, floppy disc etc. Device 22

further includes random access memory (not shown) and a communications means, such as a network interface card or other network interface means (not shown) to allow device 22 to communicate over a communication medium 26 such as the internet.

Server 24 is any suitable server, as will occur to those of skill in the art, such as the Sun Enterprise 450 server sold by Sun Microsystems of Palo Alto CA, and is generally operable to function as a network computing server. In an embodiment of the invention, server 24 is Java-based and includes a central processing unit (not shown), random access memory (not shown), a data storage means 38, and a communications means, such as a network interface card or other network interface means (not shown) to allow server 24 to communicate over communication medium 26.

Software within device 22 includes a Java-based framework 28 that is associated with Java-based objects ($O_1, O_2 \dots O_8$) which in a present embodiment are JavaBeans. Specifications for frameworks and JavaBean can be found in *Java Dynamic Management Kit 3.0 Programming Guide*, Chapter 10, 1998 Sun Microsystems, Inc. 901 San Antonio Road, Palo Alto, California 94303 U.S.A. ("JDMK") and *Sun Microsystems, JavaBeans™*, Version 1.01, Hamilton, G., Editor, 1996, Sun Microsystems Inc., Mountain View California ("*JavaBeans Specification*").

Framework 28 is associated with a client-adapter 30, which connects framework 28 to external software applications via the network interface means and over communication medium 26. It will be apparent to those of skill in the art that client-adapter 30 provides a software interface between the software and the network interface means on device 22. In a present embodiment, client-adapter 30 is the client component of an adapter determined using the adapter tool provided with the *JDMK*, and discussed in detail in Chapter 5 of *JDMK*. However, it will be apparent that other client-adapter means can be used for other types of programming, as will occur to those of skill in the art.

It will be understood by those of skill in the art that framework 28 is a type of registry for registering the existence of objects ($O_1, O_2 \dots O_8$). It will be further understood that objects ($O_1, O_2 \dots O_8$) form at least a portion of at least one software application intended for execution by a user of device 22. In other embodiments, there can be a different number of objects, and/or objects ($O_1, O_2 \dots O_8$) can be another type of

program-language or component architecture, such as ActiveX, COM or CORBA objects, as will occur to those of skill in the art. It will be further understood that a registry or framework 28 can be incorporated into each object ($O_1, O_2 \dots O_8$) thus eliminating the need for a separate framework 28.

5 Framework 28 also includes a persistent storage registry 31, that is resident in the persistent storage means of device 22, and is used to establish the contents of framework 28 on initialization or start up of device 22. Registry 31 is associated with a registry-manager object that manages the contents of registry 31. In a present embodiment registry 31 is an m-bean repository in accordance with the *JDMK* and lists objects ($O_1, O_2,$
10 $\dots O_8$) within framework 28.

Device 22 also includes a persistent storage repository 33 that is associated with a portion of the persistent storage means of device 22 and can store objects ($O_1, O_2 \dots O_8$). As shown in Figure 1, repository 33 has a plurality of storage blocks ($SB_1 \dots SB_n$), where SB_1 contains objects ($O_1, O_2 \dots O_4$), SB_2 contains objects ($O_5 \dots O_8$) and $SB_3 \dots SB_n$ are
15 empty. It is to be understood that each storage block ($SB_1 \dots SB_n$) need not contain an identical number of objects, and that the exact size of each storage block ($SB_1 \dots SB_n$) can be dynamically allocated. As such, storage blocks $SB_3 \dots SB_n$ can be considered a contiguous block of available storage space. It will be further understood that in other embodiments of the invention, certain temporary objects resident in framework 28 need
20 not be stored in repository 33.

Repository 33 is also associated with a repository-manager object that manages the storage and retrieval of data from the persistent storage means. In a present embodiment of the invention, storage blocks ($SB_1 \dots SB_n$) are Java Archives ("JAR") and repository 33 is a Java ARchive ("JAR") repository.

25 Framework 28 also includes a gateway G which in the present embodiment is a managed-bean object or "m-bean", (the specifications for which are discussed in Chapter 3 of *JDMK*). Once device 22 is initialized, gateway G has unique privileges to interact (via a method call or other suitable means) with and manage the contents of register 31 and repository 33.

Software within server 24 includes an application APP and a gateway managed-object G_{MO} , which is created by performing a managed-object generation operation on gateway G. It will be understood that gateway managed-object G_{MO} is a client-bean (“c-bean”), or proxy, shell, wrapper or other suitable representation of gateway G. As
5 gateway managed-object G_{MO} is a representation for gateway G, application APP can transparently access gateway G, through method calls to gateway managed-object G_{MO} .

In a present embodiment, the managed-object generation operation is the “mogen” tool provided with the Java Dynamic Management Kit, and discussed in “Generating a C-bean” of Chapter 2 of *Sun Microsystems, Java Dynamic Management Kit*. It is to be
10 understood, however, that other managed-object generation operations can be used as will occur to those of skill in the art.

Both application APP and gateway managed-object G_{MO} are associated with a server-adapter 36, for connection to external software applications via communication medium 26. The previously-mentioned specifications are applicable to server-adapter 36,
15 which is complementary to client-adapter 30, and collectively server-adapter 36 and client-adapter 30 compose an adapter determined using the adapter tool provided with the *JDMK*, and discussed in Chapter 5 of *JDMK*. Similar to client-adapter 30, server-adapter 36 provides a software interface between the software and the network interface means on server 24.

20 Server 24 also includes four objects (O_1 , O_5 , O_9 , O_{10}) which are stored in file storage means 38, and are intended for transfer to device 22 and execution or activation thereon. In a present embodiment objects (O_1 , O_5 , O_9 , O_{10}) are all JavaBeans, and objects O_1 and O_5 in storage means 38 are the same as objects O_1 and O_5 resident in device 22.

A method for transferring objects between server 24 and device 22 will now be
25 discussed with reference to the flowchart of Figure 2 and system 20. In a present embodiment, the method of Figure 2 is commenced based on a determination that device 22 requires objects (O_1 , O_5 , O_9 , O_{10}) for an application, and that these objects are currently stored in file storage means 38. At step 100, device 22 is interrogated to determine its configuration. In a present embodiment, this is accomplished by application
30 APP which makes a method-call to gateway G via gateway managed-object G_{MO} . During

6674790-6902260
this method-call, application APP determines the amount of persistent storage space available on device 22, and a list of objects currently stored in device 22 and/or any other properties of device 22 necessary for a file transfer. Application APP determines that SB₁ contains objects (O₁ ... O₄), SB₂ contains objects (O₅ ... O₈) and that storage block SB₃ ...
5 SB_n are available for storage.

At step 120, the objects required for transfer are determined. In a present embodiment, step 120 is accomplished by application APP, which compares the configuration data obtained at step 100 with the list of objects needed by device 22. Application APP thus determines that object O₁ and object O₅ are already resident on
10 device 22 and it is therefore only necessary to transfer object O₉ and object O₁₀ from server 24 to device 22. Application APP also notes that storage block SB₃ is empty and large enough to store object O₉ and object O₁₀.

It will be apparent that if application APP determined that no objects were required for transfer to device 22, or where there was insufficient storage blocks to store the
15 objects required for transfer, then the method would terminate and could be recommenced from step 100 at a later time, if necessary.

At step 140, the objects required by device 22 are packaged into an archive in preparation for transfer and specifically, object O₉ and object O₁₀ are packaged into an archive 40, as shown in Figure 3. In a present embodiment, archive 40 is a Java Archive
20 ("JAR") and includes a checksum or any other suitable verification means attached thereto.

At step 160, as shown in Figure 4, archive 40 is transferred from server 24 to the random access memory of device 22 via communication medium 26, as indicated in dashed line. In a present embodiment, the archive 40 is passed as a parameter within a
25 method call to gateway G.

At step 180 an archive-commit flag, which in a present embodiment is present in gateway G, is set 'on'. It will be understood that in other embodiments, the archive-commit flag can be implemented in a variety of different ways. As will be discussed in greater detail below, the archive-commit flag is used during initialization or boot-up to

determine whether an archive was being committed when the operation of device 22 was disrupted.

At step 200, as shown in Figure 5, gateway G instructs repository 33 to commit archive 40 to persistent storage. Gateway G makes a method call to the repository-
5 manager associated with repository 33 to write archive 40 to persistent storage associated with repository 33. In a present embodiment, the checksum associated with archive 40 is also used to verify the integrity of archive 40 during the storage procedure.

It is to be understood that if device 22 experiences a power failure or otherwise shuts down during step 200, then during a subsequent initialization of device 22, gateway
10 G will determine that its archive-commit flag is 'on', and determine that a power-failure or shut down occurred during step 200. Accordingly, gateway G will query the repository-manager querying as to whether archive 40 was successfully committed to storage. If successful, the method proceeds to step 210. However, if unsuccessful, the repository-manager will free-up any storage blocks being used during step 200 and, once device 22
15 reestablishes communication with server 22, application APP will then begin at step 100 and re-attempt the file transfer.

Assuming that archive 40 is successfully committed to persistent storage, as shown in Figure 5, then method of the present embodiment proceeds to step 210 where the archive-commit flag is set 'off', and a list-commit flag is set 'on'. In a present
20 embodiment, the list-commit flag is present in gateway G, and is used by gateway G during initialization or boot-up to determine whether an archive was being committed when the operation of device 22 was disrupted.

As indicated in Figure 6, at step 220 gateway G instructs registry 31 to commit the list of objects within archive 40 to persistent storage. Specifically, gateway G makes a
25 method call to the registry-manager objects associated with registry 31 to list object O₉ and object O₁₀ in the persistent storage area associated with registry 31. It is to be understood that if the operation of device 22 if device 22 experiences a power failure or otherwise shuts down then during step 220, then during the subsequent initialization of device 22, gateway G will discover that the list-commit flag is 'on', and accordingly
30 determine that step 220 failed. Accordingly, gateway G can commence a recovery

operation of step 220 by instructing registry 31 to examine the contents of repository 33 to ascertain the list of objects that should be present in registry 31, and use this information to complete step 220.

However, if the operation of device 22 is not disrupted during step 220, then object O₉ and object O₁₀ are listed within registry 31 as shown in Figure 6, and the method of the present embodiment proceeds to step 230.

At step 230, the list-commit flag is set off. At step 240, object O₉ and object O₁₀ are activated by registry 31 which instantiates object O₉ and object O₁₀ within framework 28, as shown in Figure 7. Accordingly, object O₉ and object O₁₀ now become available to applications executing on device 22.

While the embodiments discussed herein are directed to particular implementations of the present invention, it will be apparent that the sub-sets and variations to these embodiments are within the scope of the invention. For example, while the embodiments herein are directed to JavaBean objects, it will be apparent that other types of objects can be implemented in accordance with the teachings of the invention. It is also contemplated that the archive-commit flag and list-commit flags can be implemented in a variety of forms and recovery-means within device 22, in order to provide a robust mechanism to recover from a failed download and/or archive-commit procedure should device 22 lose power during such operations.

The present invention provides a novel method and system transferring objects between computers. The present invention is particularly useful in systems incorporating the Java Dynamic Management Kit, as it provides a means where only the required objects are actually transferred from the server to the device to improve the overall efficiency of the download.

Further, flags in the device can be used to determine whether a file transfer failed during initialization of the device, thereby improving the overall reliability of file transfers and providing an effective means of system recovery. File transfer reliability is further improved by the use of a verification means such as a checksum attached to the archive file before file transfer. This checksum can be used by the device to ensure the overall integrity of the objects when the objects are committed to persistent storage areas within

the device. In addition, the invention can increase efficient use of device resources by not loading redundant components on the device.

We claim:

1. A method for transferring at least a portion of a group of objects to a first
5 computer from a second computer, said first computer having a central processing unit,
random access memory and a persistent storage means, said first and second computers
connected by a communication medium, said method comprising the steps of:

interrogating said first computer for configuration information;
10 determining required objects for transfer based on an operation that considers, at
least in part, said configuration information and said group of objects;
packaging said required objects into an archive;
transferring said archive to said random access memory;
setting an archive-commit flag on;
15 committing said archive to said persistent storage;
setting said archive commit flag off;
setting a list-commit flag on;
committing a list of said required objects to said persistent storage;
setting said list-commit flag off; and
20 activating said required objects in said random access memory.

2. The method according to claim 1 wherein said objects are JavaBeans beans based
on the JavaBeans specification.

3. The method according to claim 1 wherein said archive is a Java ARchive.

4. The method according to claim 1 wherein said archive is committed to a repository
area of said persistent storage.

5. The method according to claim 1 wherein said list is committed to a registry area of said persistent storage.

6. A method in the initializing a device comprising the steps of:

determining whether an archive was being committed to persistent storage when said device was powered-off; and

instructing said persistent storage to clear the portion of said archive committed to persistent storage when said step of determining establishes that an archive was being committed when said device was powered-off.

7. The method according to claim 4 wherein said archive is a Java Archive of JavaBean objects.

8. The method according to claim 1 wherein said archive is committed to a repository area of said persistent storage.

9. A method in the initialization of a device comprising the steps of:

determining whether a list of objects from an archive was being committed to a registry area of persistent storage when said device was powered-off; and

examining said archive to determine a remaining list of objects to be committed to said registry area when said step of determining establishes that a list was being committed when said device was powered-off; and

committing the remainder of said list established from said examining step.

10. The method according to claim 6 wherein said archive is a Java Archive of JavaBean objects.

11. The method according to claim 6 wherein said archive is committed to a repository area of said persistent storage.

12. A system for transferring at least a portion of a group of objects between computers, comprising:

5 a first computer having a central processing unit, random access memory, a persistent storage repository for storing archives of objects, a persistent storage registry for storing a list of said objects, and a first network interface means, said first computer operable to execute said objects, and operable to execute a gateway object having privileges to interact with said repository and said register;

10 a second computer having a second interface means connectable to said first computer via a communication medium, said second computer having a file storage means for storing a group of objects, said second computer being operable to execute an application and a gateway managed object, said gateway managed object being a proxy for
15 said gateway object so that said application can interrogate said gateway object for configuration information of said device and transfer an archive of at least a portion of said group via said gateway managed object to said gateway object, said archive packaged based on an operation that considers, at least in part, said configuration information;

20 such that when said gateway object receives said archive, said gateway object instructs said repository to commit said archive to said repository and a list of objects in said archive to said registry, said device setting at least one flag during said commission of said archive and said list to allow device recovery upon initialization if said device loses power during said commission.

25 13. The system according to claim 12 wherein said objects are JavaBeans beans based on the JavaBeans specification.

14. The system according to claim 12 wherein said archive is a Java ARchive.

15. The system according to claim 12 wherein said configuration information includes an inventory of available random access memory on said device.

5 16. The system according to claim 12 wherein said configuration information includes an inventory of available persistent storage on said device.

17. The system according to claim 12 wherein said configuration information includes a list of objects from said group of objects which are already resident in said device.

10 18. The system according to claim 12 wherein said gateway managed object is determined using the mogen tool from the Java Dynamic Management Kit.

19. The system according to claim 12 wherein said first computer is a Java-based intelligent device and said second computer is a Java-based network computing server.

15

20. The system according to claim 12 including a first flag set during said commission of said archive and a second flag set during said commission of said list.

ABSTRACT

A method and system for transferring objects between computers, especially a server and a device in a Java environment, is provided. A server connected to the device
5 interrogates the device to determine whether the device has the available resources to accept a download, and to determine if any of objects included in the download are already resident on the device. Using a gateway object on the device, and a gateway managed-object on the server, the server passes an archive of the necessary objects, not presently available on the device, to the gateway via the gateway managed-object. The
10 gateway then instructs a persistent storage repository on the device to commit the archive to persistent storage. A flag is set at the start of the archive-committing process, such that if the device loses power during the archive-committing process, upon re-initialization of the device, the device can determine that the archive-committing process failed and will instruct the persistent storage to free-up storage area used during the archive-committing
15 process. The transferal of objects can then be recommenced once the server and device reestablish communication. The present invention can provide an effective means to ensure a stable recovery of the device upon re-initialization, should the file transfer and/or archive-commit process fail due to a power failure to the device. Further, the invention can increase efficient use of device resources by not loading redundant components on the
20 device and can reduce bandwidth requirements and/or download times as redundant information is not transferred.

Fig. 1

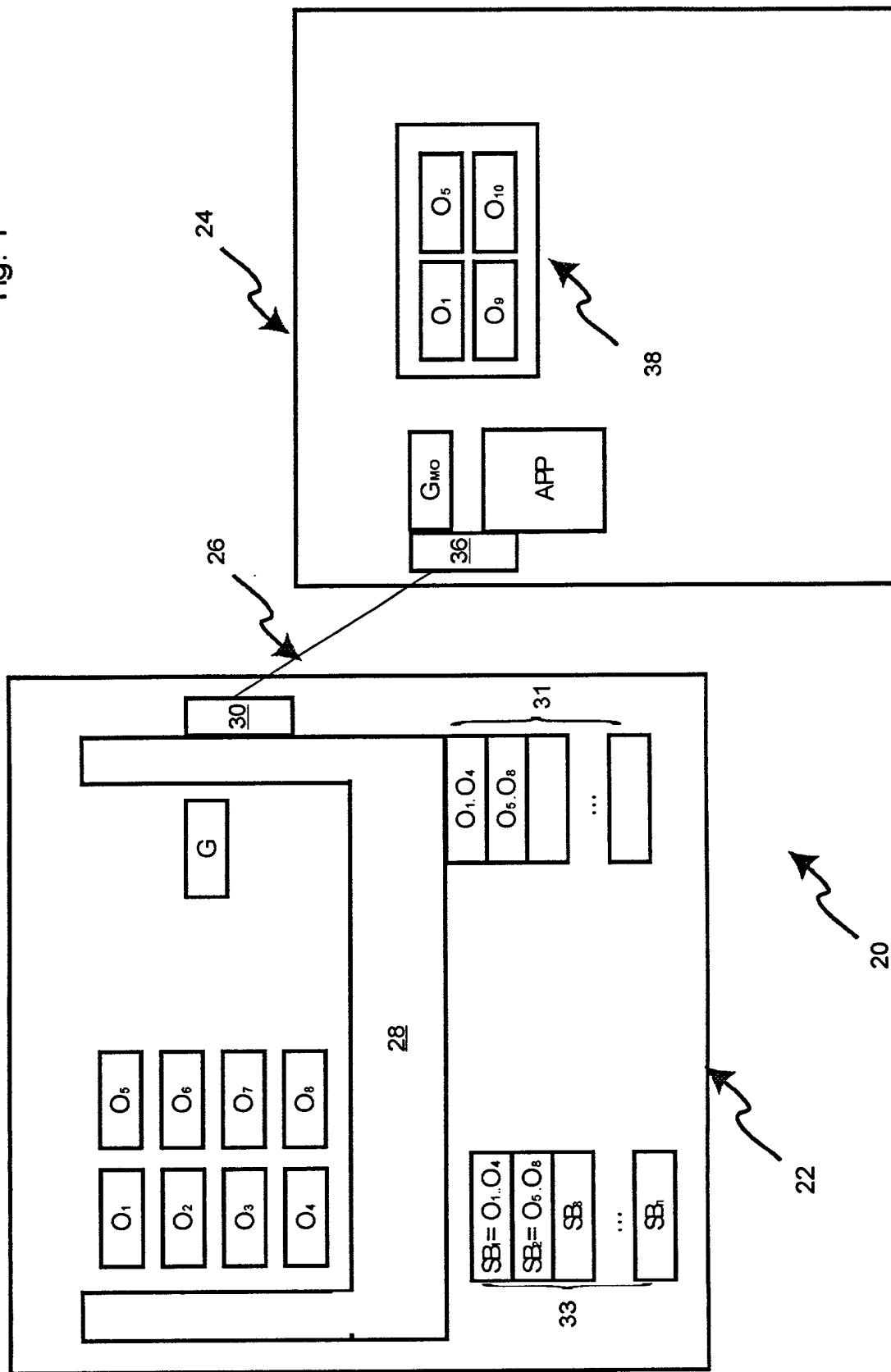


Fig. 2

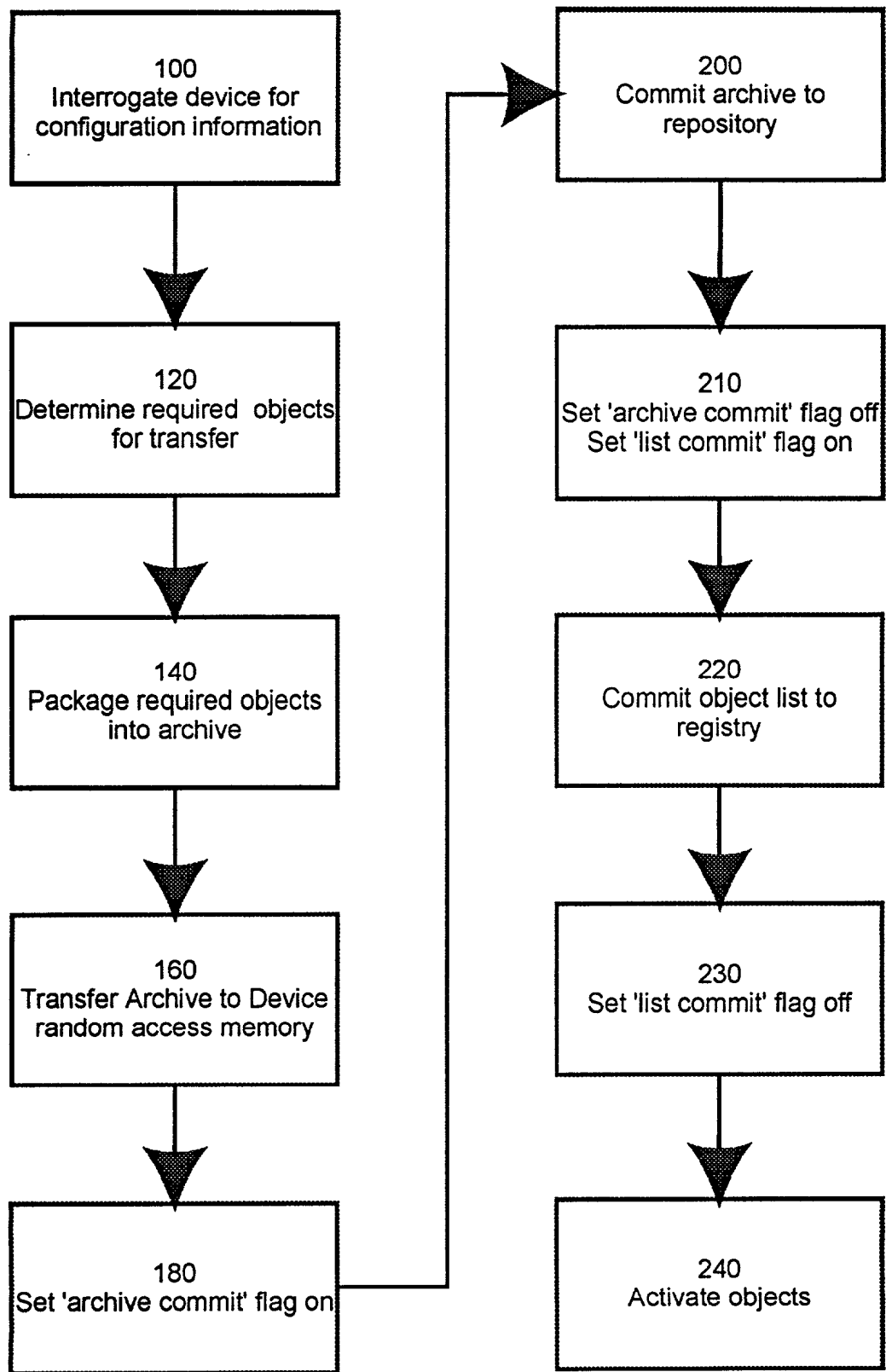


Fig. 3

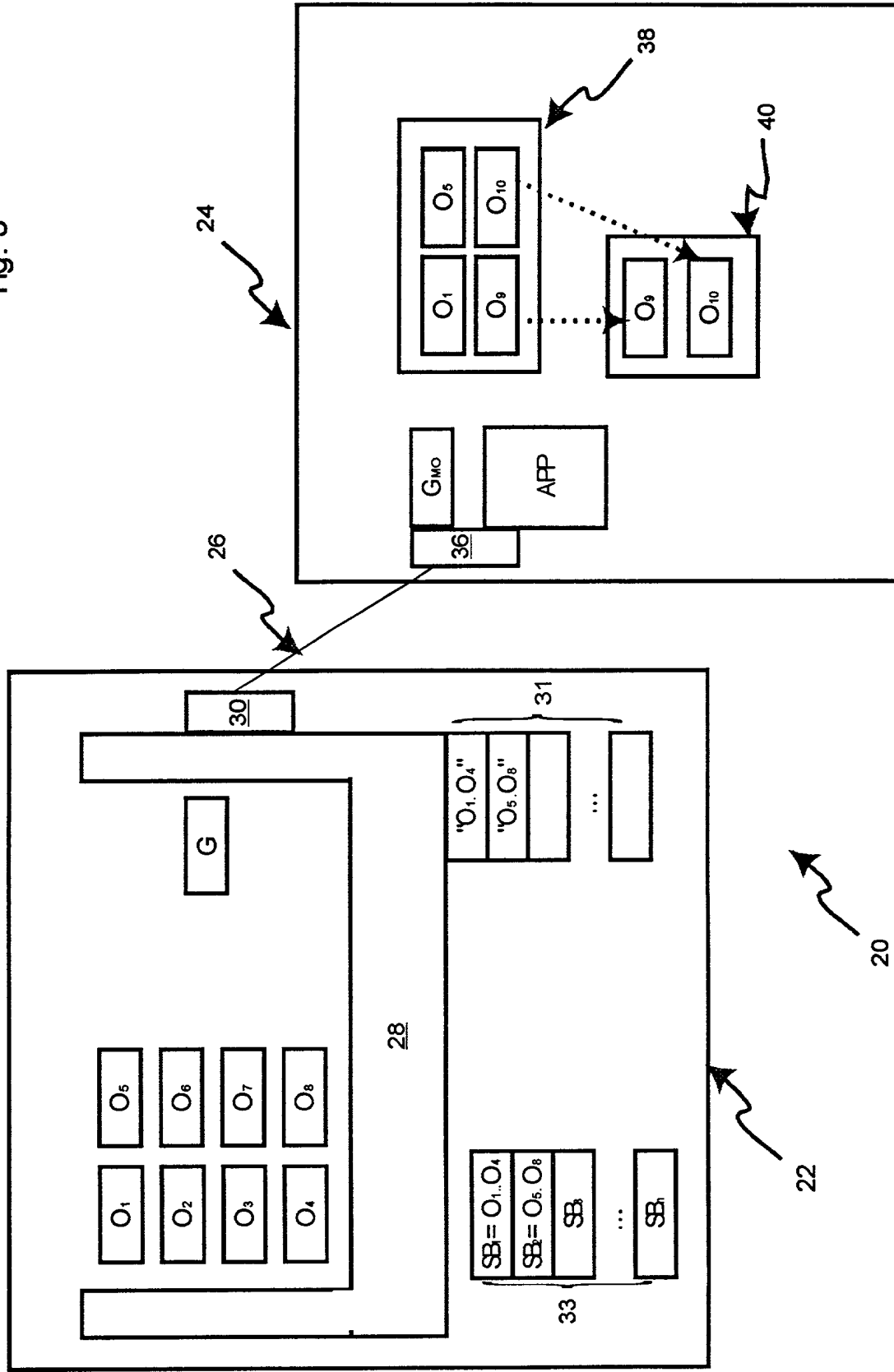


Fig. 4

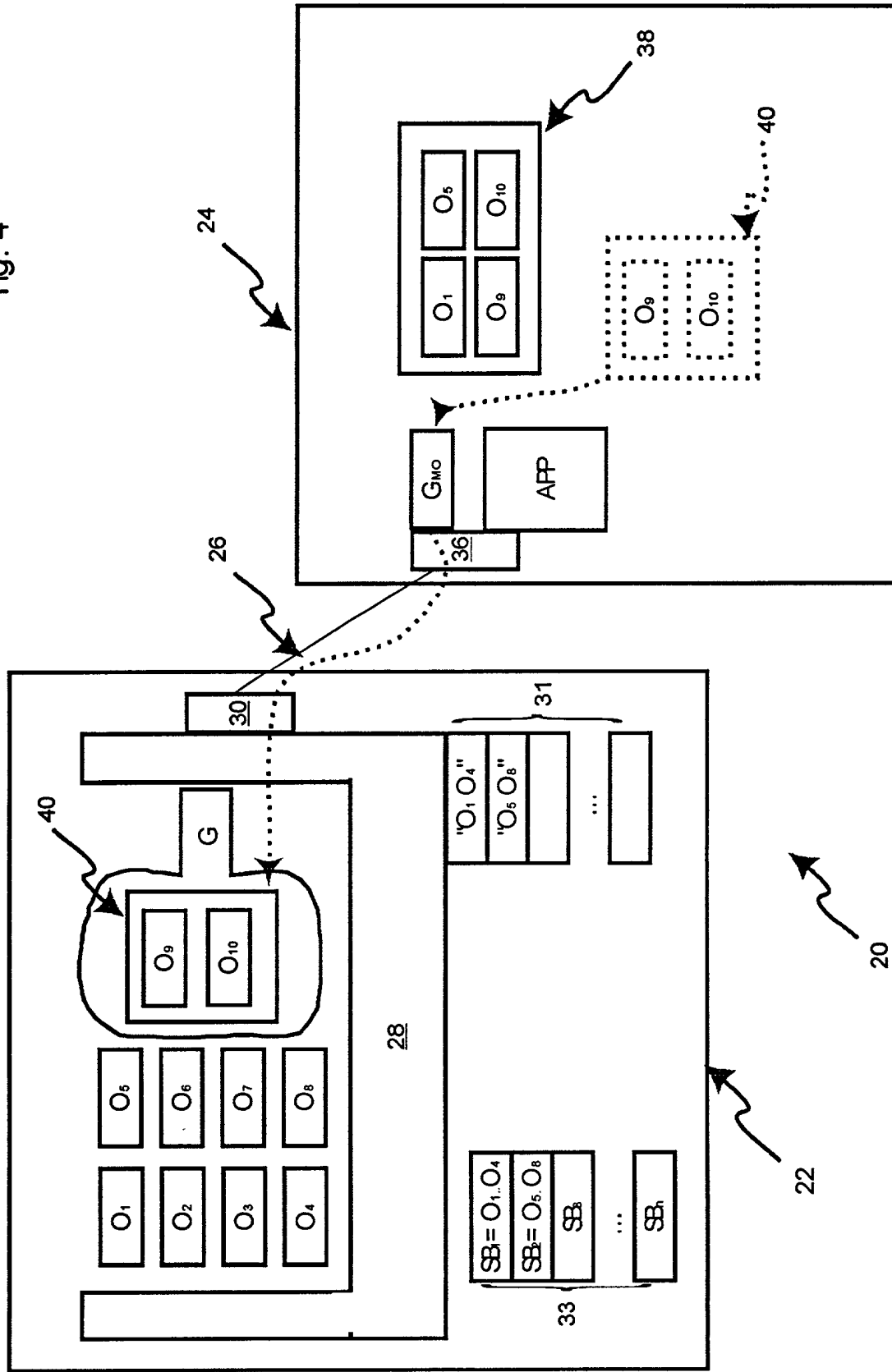


Fig. 5

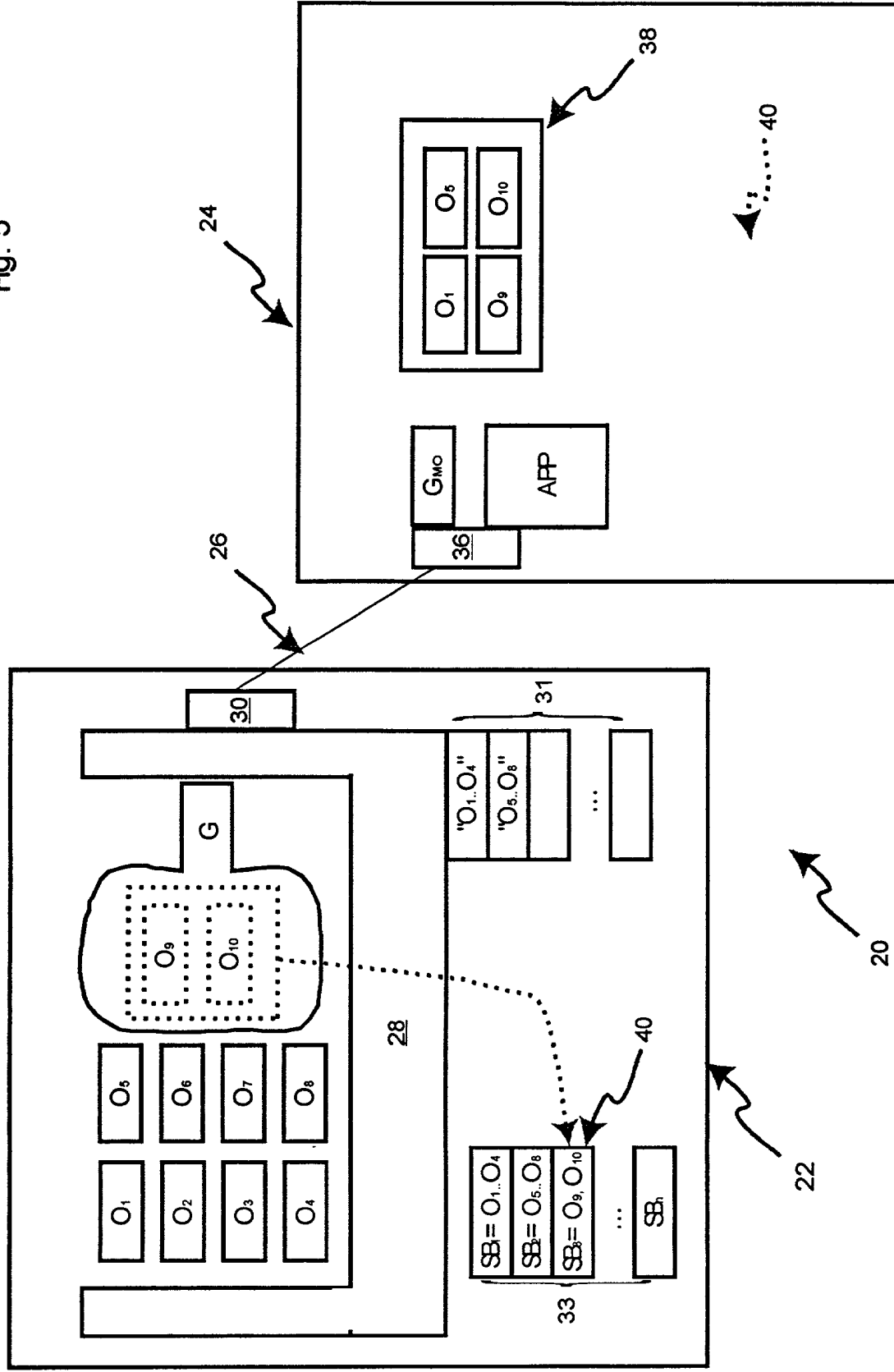


Fig. 6

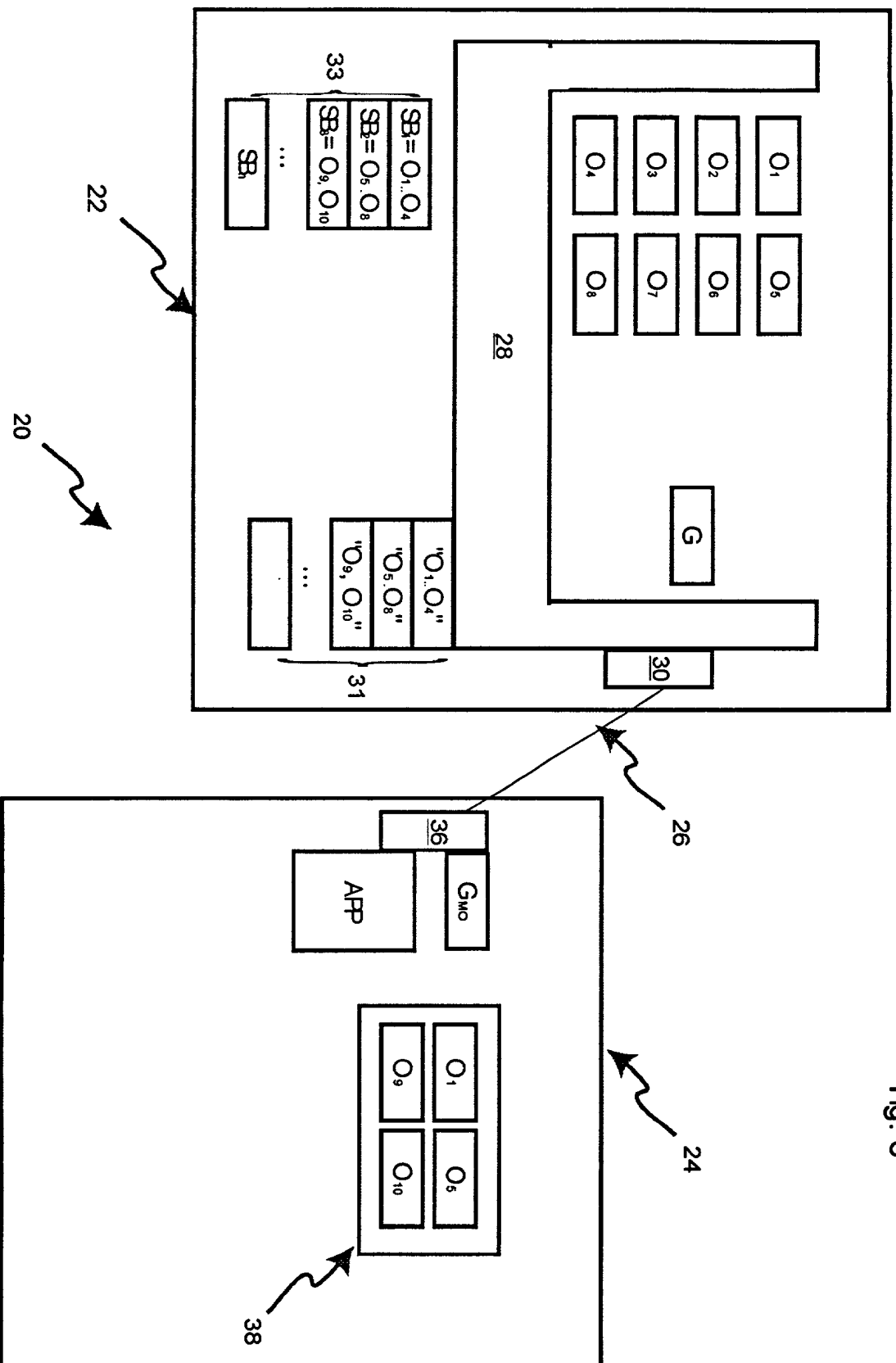
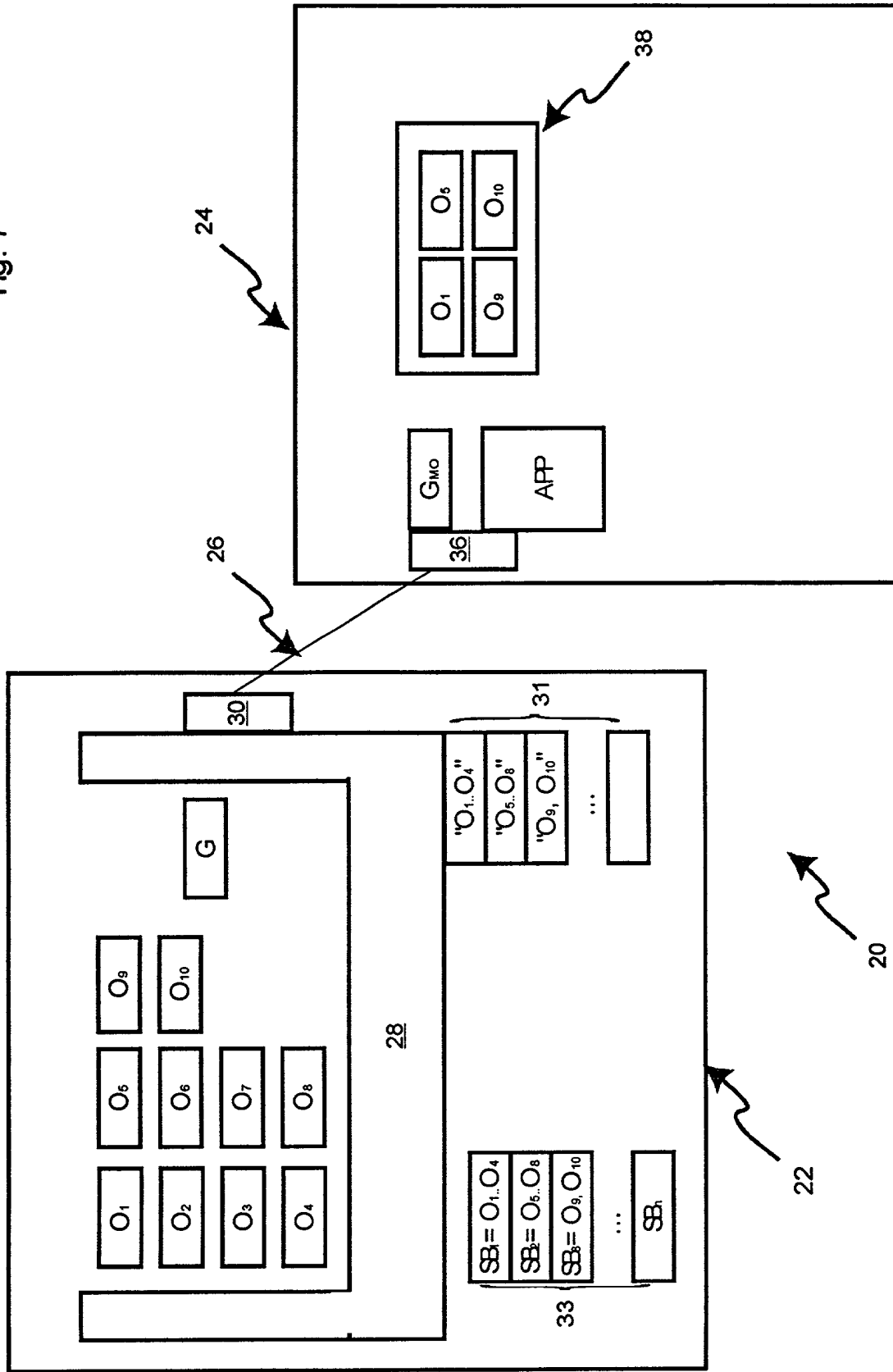


Fig. 7



Docket No.

T8-465038US

Declaration and Power of Attorney For Patent Application

English Language Declaration

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

METHOD AND APPARATUS FOR INCREMENTAL DOWNLOAD FROM SERVER TO CLIENT

the specification of which

(check one)

☒ is attached hereto.

☐ was filed on _____ as United States Application No. or PCT International Application Number _____ and was amended on _____ (if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d) or Section 365(b) of any foreign application(s) for patent or inventor's certificate, or Section 365(a) of any PCT International application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate or PCT International application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)

Priority Not Claimed

(Number)

(Country)

(Day/Month/Year Filed)

☐

(Number)

(Country)

(Day/Month/Year Filed)

☐

(Number)

(Country)

(Day/Month/Year Filed)

☐

I hereby claim the benefit under 35 U.S.C. Section 119(e) of any United States provisional application(s) listed below:

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

(Application Serial No.)

(Filing Date)

I hereby claim the benefit under 35 U. S. C. Section 120 of any United States application(s), or Section 365(c) of any PCT International application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT International application in the manner provided by the first paragraph of 35 U.S.C. Section 112, I acknowledge the duty to disclose to the United States Patent and Trademark Office all information known to me to be material to patentability as defined in Title 37, C. F. R., Section 1.56 which became available between the filing date of the prior application and the national or PCT International filing date of this application:

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

(Application Serial No.)

(Filing Date)

(Status)
(patented, pending, abandoned)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. *(list name and registration number)*

Richard P. Bauer, Registration No. 31,588

Send Correspondence to:

**Fitzpatrick, Cella, Harper & Scinto
30 Rockefeller Plaza
New York, NY 10112-3801 U.S.A.**

Direct Telephone Calls to: *(name and telephone number)*

Richard P. Bauer (Tel: (202) 721-5444)

Full name of sole or first inventor	STEPHEN MARYKA	
Sole or first inventor's signature	<i>[Signature]</i>	Date <i>x June 14, 1999</i>
Residence	292 HAWKWOOD DRIVE N.W. CALGARY, ALBERTA T3G 3N9 CANADA	
Citizenship	CANADIAN	
Post Office Address	SAME AS ABOVE	

Full name of second inventor, if any	BERTRAND MICHAUD	
Second inventor's signature	<i>[Signature]</i>	Date <i>x June 14, 1999</i>
Residence	27 BEDDINGTON GREEN N.W. CALGARY, ALBERTA T3K 1M7 CANADA	
Citizenship	CANADIAN	
Post Office Address	SAME AS ABOVE	

Full name of third inventor, if any	
GREG DICK	
Third inventor's signature	
X	<i>Gregory J. Dick</i>
Residence	Date
1701 BROADVIEW ROAD N.W. CALGARY, AB T2W 3H4 CANADA	X 14 JUN 1999
Citizenship	
CANADIAN	
Post Office Address	
SAME AS ABOVE	

Full name of fourth inventor, if any	
Fourth inventor's signature	
Residence	Date
Citizenship	
Post Office Address	

Full name of fifth inventor, if any	
Fifth inventor's signature	
Residence	Date
Citizenship	
Post Office Address	

Full name of sixth inventor, if any	
Sixth inventor's signature	
Residence	Date
Citizenship	
Post Office Address	